Our Ref. No. 042390.P10578
Express Mail No.: EL802874799

UNITED STATES PATENT APPLICATION

FOR

DYNAMIC DELAYED TRANSACTION BUFFER CONFIGURATION BASED ON BUS FREQUENCY

INVENTOR:

Mikal C. Hunsaker

PREPARED BY:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 Wilshire Blvd., 7th Floor
Los Angeles, CA 90025-1026
(714) 557-3800

BACKGROUND

1. Field of the Invention

[0001] This invention relates to buffer circuits. In particular, the invention relates to dynamically configured buffer circuits.

2. Description of Related Art

[0002] Due to the long latency for a Peripheral Component Interconnect (PCI) or PCI-X read request from an input/ output (I/O) device, a PCI/ PCI-X bridge generally treats the inbound read transaction as a delayed transaction in PCI mode or a split transaction in PCI-X mode. A PCI delayed transaction occurs when the I/O device issues an initial read request. When the PCI bridge receives the read completion data, it stores in it in a delayed transaction buffer. When the PCI master returns with the same read request, the read completion data is provided from the delayed transaction buffer. Similarly, a PCI-X split transaction occurs when the I/O device issues an initial read request. Since the PCI-X bridge does not have the read data, it terminates the transaction with a Split Response indicating that the bridge has accepted the read request and will later provide the I/O device with the read completion data. When the bridge receives the read completion data, it stores it in a delayed transaction buffer. The bridge then sends the data to the I/O device as a Split Completion transaction.

[0003] A PCI/ PCI-X bridge may support multiple concurrent delayed transactions if there are multiple I./O devices or a single I/O device with multiple read requests. In addition, the bus frequency may vary depending on system configurations. The PCI bus can operate at frequencies of 33 MHz or 66 MHz.

The PCI-X bus can operate at frequencies of 66MHz, 100 MHz, or 133 MHz. At these various frequencies, the number of read delayed transactions may be different which affect the depth of the delayed transaction buffer. Existing techniques for buffer management are inefficient. One technique uses a single large content addressable memory (CAM) delayed transaction buffer to support multiple delayed transactions. This technique requires a complex buffer management scheme and is costly. Another technique uses multiple delayed transaction buffers that are deep enough to handle the highest PCI/ PCI-X bandwidth and lowest system latency. This technique requires high gate count and wastes hardware when used with lower bus frequencies.

[0004] Therefore, there is a need to have an efficient technique to handle multiple delayed transactions for various bus frequencies.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] The features and advantages of the present invention will become apparent from the following detailed description of the present invention in which:

[0006] Figure 1 is a diagram illustrating a system in which one embodiment of the invention can be practiced.

[0007] Figure 2 is a diagram illustrating a buffer circuit shown in Figure 1 according to one embodiment of the invention.

[0008] Figure 3A is a diagram illustrating a buffer circuit configured for low bus frequencies according to one embodiment of the invention.

[0009] Figure 3B is a diagram illustrating a buffer circuit configured for high bus frequencies according to one embodiment of the invention.

[0010] Figure 4 is a flowchart illustrating a process to Delayed Transaction (DT) data buffer according to one embodiment of the invention.

DESCRIPTION

[0011] In the following description, for purposes of explanation, numerous details are set forth in order to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that these specific details are not required in order to practice the present invention. In other instances, well-known electrical structures and circuits are shown in block diagram form in order not to obscure the present invention.

[0012] Figure 1 is a diagram illustrating a computer system 100 in which one embodiment of the invention can be practiced. The computer system 100 includes a processor 110, a host bus 120, a memory control hub (MCH) 130, a system memory 140, an input/output control hub (ICH) 150, a mass storage device 170, and input/output devices 180_1 to 180_K and a peripheral bridge 190.

[0013] The processor 110 represents a central processing unit of any type of architecture, such as embedded processors, micro-controllers, digital signal processors, superscalar computers, vector processors, single instruction multiple data (SIMD) computers, complex instruction set computers (CISC), reduced instruction set computers (RISC), very long instruction word (VLIW), or hybrid architecture. In one embodiment, the processor 110 is compatible with the Intel Architecture (IA) processor, such as the IA-32 and the IA-64. The processor 110 typical contains a number of control registers to support memory management tasks such as virtual memory and cache memory. These tasks may include paging and segmentation.

[0014] The host bus 120 provides interface signals to allow the processor 110 to communicate with other processors or devices, e.g., the MCH 130. The host bus 120 may support a uni-processor or multiprocessor configuration. The

host bus 120 may be parallel, sequential, pipelined, asynchronous, synchronous, or any combination thereof.

[0015] The MCH 130 provides control and configuration of memory and input/output devices such as the system memory 140, the ICH 150, and the peripheral bridge 190. The MCH 130 may be integrated into a chipset that integrates multiple functionalities such as the isolated execution mode, host-to-peripheral bus interface, memory control. For clarity, not all the peripheral buses are shown. It is contemplated that the system 100 may also include peripheral buses such as Peripheral Component Interconnect (PCI), accelerated graphics port (AGP), Industry Standard Architecture (ISA) bus, and Universal Serial Bus (USB), etc.

[0016] The system memory 140 stores system code and data. The system memory 140 is typically implemented with dynamic random access memory (DRAM) or static random access memory (SRAM). The system memory 140 may also include other programs or data which are not shown, such as an operating system.

[0017] The ICH 150 has a number of functionalities that are designed to support I/O functions. The ICH 150 may also be integrated into a chipset together or separate from the MCH 130 to perform I/O functions. The ICH 150 may include a number of interface and I/O functions such as PCI bus interface, processor interface, interrupt controller, direct memory access (DMA) controller, power management logic, timer, universal serial bus (USB) interface, mass storage interface, low pin count (LPC) interface, etc. The ICH 150 has interface to the PCI bus 160. The PCI bus is coupled to M PCI devices 165_1 to 165_M.

[0018] The mass storage device 170 stores archive information such as code, programs, files, data, applications, and operating systems. The mass storage device 170 may include compact disk (CD) ROM 172, floppy diskettes 174, and hard drive 176, and any other magnetic or optic storage devices. The mass storage device 170 provides a mechanism to read machine-readable media.

[0019] The I/O devices 180_1 to 180_K may include any I/O devices to perform I/O functions. Examples of I/O devices 180_1 to 180_K include controller for input devices (e.g., keyboard, mouse, trackball, pointing device), media card (e.g., audio, video, graphics), network card, and any other peripheral controllers.

[0020] The peripheral bridge 190 connects to the MCH 130 over a primary bus 195. The peripheral bridge is coupled to P PCI/PCI-X devices 187_1 through 187_p through a secondary peripheral PCI/PCI-X bus 185. The secondary peripheral bus 185 can operate at frequencies of 33MHz, 66MHz, 100MHz, and 133 MHz. The peripheral bridge 190 contains a delayed/split transaction (DT) buffer circuit 192 for servicing PCI/PCI-X reads.

[0021] The DT data may be requested by a delayed transaction read request from the PCI/ PCI-X devices 187_1 to 187p. The request may result in a delayed transaction (for PCI bus) or a split response (for PCI-X bus). The peripheral bridge 190 receives the read request from the PCI/ PCI-X devices. The peripheral bridge 190 may not have the requested data and therefore returns an acknowledgment to the requesting device. Then, when the peripheral bridge 190 fetches the DT data from the memory 140, it stores the DT data in the buffer circuit 192 to be transferred to the PCI/ PCI-X device via the PCI/ PCI-X bus 185. Since the PCI/ PCI-X bus may operate at one of several bus frequencies (e.g., 33 MHz, 66 MHz, 100 MHz, and 133 MHz), the number of read requests and the

amount of requests may differ for various frequencies. The buffer circuit 135 is dynamically configured to accommodate the various frequencies.

[0022] It is noted that the invention may be described as a process which is usually depicted as a flowchart, a flow diagram, a structure diagram, or a block diagram. Although a flowchart may describe the operations as a sequential process, many of the operations can be performed in parallel or concurrently. In addition, the order of the operations may be re-arranged. A process is terminated when its operations are completed. A process may correspond to a method, a function, a procedure, a subroutine, a subprogram, etc. When a process corresponds to a function, its termination corresponds to a return of the function to the calling function or the main function.

[0023] Figure 2 is a diagram illustrating the buffer circuit 192 shown in Figure 1 according to one embodiment of the invention. The buffer circuit 192 includes a data steering circuit 210 and N delayed transaction (DT) buffers 220_1 to 220_N.

[0024] The data steering circuit 210 provides the input and output data paths to the primary bus 195 as input and the peripheral secondary PCI/PCI-X bus 185 as output. The data steering circuit 210 is dynamically configured according to the frequency of the peripheral secondary bus 185. In addition, although the invention is described in terms of a delayed transaction read request from the I./O device to the memory, the concept can be readily applied or extended to similar situations where a dynamically configured buffering is needed. This includes a buffered write, a buffered read, etc. The data steering circuit 210 includes a control circuit 212, a dynamically configured input circuit 214, and a dynamically configured output circuit 216.

[0025] The control circuit 212 is a logic circuit that receives the bus frequency information and generates appropriate control signals to the input and output circuits 214 and 216. The control signals may include configuration bits and selection signals to the de-multiplexing and/or multiplexing circuits in the input and output circuits 214 and 216.

[0026] The input circuit 214 is interface primary bus 195 and the N DT buffers 220_1 to 220_N at the input ports of these buffers. The input circuit 214 transfers a delayed transaction (DT) data having a transaction identifier to one of the N DT buffers 220_1 to 220_N which is associated with that transaction identifier. The input circuit 214 is dynamically configured according to the bus frequency. The transaction identifier may be the address of the DT data, or any identifier employed by the chipset 190 to identify the transactions.

[0027] The output circuit 216 is interface to the secondary peripheral bus 185 and the N DT buffers 220_1 to 220_N at the output ports of these buffers. The output circuit 216 transfers the delayed transaction (DT) data having a transaction identifier from one of the N DT buffers 220_1 to 220_N which is associated with that transaction identifier to the secondary peripheral bus 185 to be received by the peripheral device that generates the read request. The output circuit 216 is dynamically configured according to the bus frequency. The transaction identifier may be the address of the DT data, or any identifier employed by the chipset 130 to identify the transactions.

[0028] The operating bus frequency may be low, medium, or high depending on the system design and configurations. The buffer circuit 192 may have a number of configurations to accommodate these frequencies. At lower frequencies (e.g., PCI bus at 33 and 66 MHz), the peripheral bus 185 may be able

to support many PCI devices. This results in a need to support many concurrent DTs. In addition, since these DTs are from lower frequencies, large DT buffer depths are not required. On the other hand, at high bus frequencies (e.g., PCI-X bus at 100 MHz and 133 MHz), the peripheral bus may be only able to support fewer PCI devices. In addition, large DT buffer depths are necessary to buffer more data due to higher bus bandwidth. This results in fewer DT buffers, but each buffer has larger depth.

[0029] To configure the buffers to have large depth, the buffers can be grouped together and are accessed as if they form into a single contiguous buffer. For example, suppose there are N buffers, each having a depth of D, i.e., each buffer can store up to D data items. These buffers are then configured for the lowest bus frequencies where the number of buffers is the highest and the buffer depth is lowest. At higher frequencies, the N buffers may be divided into groups. For example, suppose it is desired to increase the buffer depth to 2*D and reduce the number of buffers to N/2, then N buffers may be grouped to form N/2 groups, each group having 2 buffers. The two buffers in each group are accessed in a ping-pong or alternate manner such that they operate like a single buffer having a depth of 2*D. In general, the N buffers may be formed into P groups, each group having Q buffers where $Q = N/P$ and N, P, and Q are positive integers with $P < N$. The buffer circuit 192 may be dynamically configured to support any number of configurations. In one embodiment, there are two configurations: lower and higher bus frequencies. Figures 3A and 3B provide examples when $N = 4$.

[0030] Figure 3A is a diagram illustrating the buffer circuit 192 configured for lower bus frequencies according to one embodiment of the invention. In this example, there are four DT buffers 220_1 to 220_4 ($N = 4$). The

input circuit 214 includes a one-to-four demultiplexer (de-mux) 314. The output circuit 216 includes a four-to-one multiplexer (mux) 316.

[0031] The one-to-four de-mux 314 receives the DT data from the primary bus 195. Based on the transaction identifier, the control circuit 212 generates select signals to the de-mux 314 so that the data can be routed to the corresponding DT buffer. In addition, the control circuit 212 may also generate control signals to the corresponding DT buffer for a write or store operation.

[0032] When it is time to retrieve the DT data, the four-to-one mux 316 transfers the DT data from the DT buffer associated with the desired transaction identifier to the secondary peripheral bus 185 as requested. The control circuit 212 generates the control signals to the corresponding DT buffer for a read or load operation. In addition, the control circuit 212 generates select signals to the mux 316 so that the DT data read from the corresponding DT buffer can be routed to the secondary peripheral bus 185.

[0033] Figure 3B is a diagram illustrating the buffer circuit 192 configured for higher bus frequencies according to one embodiment of the invention. In this example, there are four DT buffers 220_1 to 220_4 (N = 4), and there are two groups (P = 2), each group having two buffers (Q = 2).

[0034] The input circuit 214 includes a one-to-two de-mux 322 and two one-to-two de-muxes 324 and 326. The output circuit 216 includes two two-to-one muxes 332 and 334 and a two-to-one mux 336. In general, when there are P groups, each group having Q buffers, the input circuit 214 includes a 1-to-P de-multiplexer and P 1-to-Q de-multiplexers. The 1-to-P de-multiplexer transfers the DT data to one of P signal paths. Each of the 1-to-Q de-multiplexers is coupled to

Q of the N buffers to transfer the DT data to one of the Q buffers based on the transaction identifier. Similarly, the output circuit 216 includes P Q-to-1 multiplexers and a P-to-1 multiplexer. The P Q-to-1 multiplexers are coupled to Q of the N buffers to transfer the DT data from one of the Q buffers to P signal paths based on the transaction identifier. The P-to-1 multiplexer is coupled to the P Q-to-1 multiplexers via the P signal paths to transfer the DT data to the secondary peripheral bus 185.

[0034] The DT buffers 220_1 to 220_4 are formed into two groups 310_1 and 310_2, each group having two buffers. The group 310_1 includes the DT buffers 220_1 and 220_2 and is interfaced to the one-to-two de-mux 324. The group 310_2 includes the DT buffers 220_3 and 220_4 and is interfaced to the one-to-two de-mux 326. In this configuration, there are in essence two buffers and each buffer has twice the depth than that of Figure 3A. The one-to-two de-mux 322 is controlled by the control circuit 212 to transfer the DT data to one of the two one-to-two de-muxes 324 and 326. Each of the one-to-two de-muxes 324 and 326 operates in a ping-pong or alternate manner to transfer the DT data to the proper DT buffer such that the DT buffer is not overflowed. For example, if the group 3101 is to store the DT data, then the one-to-two 324 de-mux stores the DT data into one of the buffers initially, say, DT buffer 1 220_1. When the DT buffer 1 220_1 is filled up, the one-to-two de-mux 324 switches to the DT buffer 2 220_2. The DT buffer 1 220_1 is available for reading. When the DT buffer 2 220_2 is filled up, presumably by that time the DT buffer 1 220_1 has be read at least partially, the one-to-two de-mux 324 switches back to the DT buffer 1 220_1. To avoid overflow the buffers, the grouping and the depth of the buffers are selected in advance based on the maximum bus frequency and the statistics of the read requests from the I/O devices.

[0035] Similarly, when the DT data is read from the buffers, the control circuit 212 generates control signals to the output circuit 216 and the DT buffers. The two-to-one mux 336 selects the data from one of the two two-to-one muxes 332 and 334 according to the transaction identifier. The selected one of the muxes 332 and 334 then reads the corresponding group of buffers in a ping-pong or alternate manner. In other words, when one of the buffer is empty, indicating that all data have been read, the selected mux then reads the other buffer, presumably by that time has been filled with DT data at least partially.

[0036] Figure 4 is a flowchart illustrating a process 400 to buffer DT data according to one embodiment of the invention.

[0037] Upon START, the process 400 configures the buffer circuit according to the bus frequency (Block 410). The configuration may be performed by writing a configuration word in a register which sets up the control circuit. Then, the process 400 receives a read request from the I/O device (Block 420). Next, the chipset bridge accepts the request and attempting to fetch data from the memory (Block 430). Then, the process 400 receives the DT data from the memory or host bus with a transaction identifier K (Block 440).

[0038] Next, the process 400 switches the de-multiplexing circuit in the input circuit dynamically according to the bus frequency (Block 450). Then, the process 400 stores the DT data to the DT buffer associated with the transaction identifier (Block 460). When there is time to read the DT data, or when the read request for data returns (Block 470), the process 400 switches the multiplexing circuit in the output circuit according to the bus frequency (Block 480). Next, the process 400 transfers the DT data from the DT buffer associated with the transaction identifier K to the peripheral bus and is then terminated.

[0039] While this invention has been described with reference to illustrative embodiments, this description is not intended to be construed in a limiting sense. Various modifications of the illustrative embodiments, as well as other embodiments of the invention, which are apparent to persons skilled in the art to which the invention pertains are deemed to lie within the spirit and scope of the invention.